

Contents

CONTEXT SR1_Path_ctx	2
CONTEXT SR2_Block_ctx	3
CONTEXT SR3_Point_ctx	4
CONTEXT SR4_TVD_ctx	5
CONTEXT SR5_Signal_ctx	6
MACHINE SR_M1	7
MACHINE SR_M2	9
MACHINE SR_M3	11
MACHINE SR_M4	13
MACHINE SR_M5	16

CONTEXT SR1_Path_ctx**SETS**

PATH

RouteId

CONSTANTS

PathConflict

Route2InitPath

PathSub

NullPath

AXIOMS**assocMultPathconflic:** $PathConflict \in PATH \leftrightarrow PATH$ **axm1:** $\forall p.p \in PATH \Rightarrow (PathConflict \cap id = \emptyset)$ **axm2:** $\forall p.p \in PATH \Rightarrow (PathConflict = PathConflict^{-1})$ **axm3:** $\forall p.p \in PATH \Rightarrow (finite(PATH))$ **assocRoute2Path:** $Route2InitPath \in RouteId \rightarrow PATH$ **assocPath_SubPath:** $PathSub \in PATH \leftrightarrow PATH$ **axm4:** $\forall p.p \in PATH \Rightarrow (\forall p1, p2.p1 \in PathSub[\{p2\}] \wedge p \notin PathConflict[\{p2\}] \Rightarrow p \notin PathConflict[\{p1\}])$ **axm5:** $\forall p.p \in PATH \Rightarrow NullPath \in PathSub[\{p\}]$ **axm6:** $NullPath \in PATH$ **END**

CONTEXT SR2_Block_ctx

EXTENDS SR1_Path_ctx

SETS

BLOCK

CONSTANTS

Path2Block

PathReduce

AXIOMS

axm1: $Path2Block \in PATH \leftrightarrow BLOCK$

axm2: $\forall p \cdot p \in PATH \Rightarrow (\forall q \cdot q \notin PathConflict[\{p\}] \Leftrightarrow (Path2Block[\{p\}] \cap Path2Block[\{q\}] = \emptyset))$

axm3: $PathReduce \in \{PATH \setminus \{NullPath\}\} \rightarrow (BLOCK \rightarrow PATH)$

axm4: $\forall p \cdot p \in PATH \setminus \{NullPath\} \Rightarrow (\exists b \cdot b \in BLOCK \Rightarrow PathReduce(\{p\})(b) \in PathSub[\{p\}])$

axm5: $\forall p \cdot p \in PATH \setminus \{NullPath\} \Rightarrow (\exists b \cdot b \in BLOCK \Rightarrow Path2Block[\{PathReduce(\{p\})(b)\}] = Path2Block[\{p\}] \setminus \{b\})$

axm6: $Path2Block[\{NullPath\}] = \emptyset$

END

CONTEXT SR3.Point_ctx

EXTENDS SR2.Block_ctx

SETS

POS

CONSTANTS

POINT

Default_Point2Pos

Route_Point2Pos

AXIOMS

axm1: $POINT \subseteq BLOCK$

axm2: $Route_Point2Pos \in RouteId \rightarrow (POINT \rightarrow POS)$

axm4: $Default_Point2Pos \in POINT \rightarrow POS$

axm3: $\forall pos \cdot pos \in POS \Rightarrow (finite(POS))$

END

CONTEXT SR4_TVD.ctx

EXTENDS SR3_Point.ctx

SETS

STATUS

CONSTANTS

Vacant

Occupied

AXIOMS

axm1: $partition(STATUS, \{Vacant\}, \{Occupied\})$

END

CONTEXT SR5_Signal_ctx

EXTENDS SR4_TVD_ctx

SETS

ASPECT

CONSTANTS

Go

Stop

AXIOMS

axm1: $\text{partition}(\text{ASPECT}, \{\text{Go}\}, \{\text{Stop}\})$

END

MACHINE SR_M1**SEES** SR1_Path.ctx**VARIABLES**

RouteId_Res
 RouteId_Occ
 Route2CurrPath

INVARIANTS

typeof RouteId_Res: $RouteId_Res \subseteq RouteId$
typeof RouteId_Occ: $RouteId_Occ \subseteq RouteId$
safety_Req: $\forall r1, r2. (r1 \neq r2 \wedge r1 \in RouteId_Occ \wedge r2 \in RouteId_Occ) \Rightarrow (PathConflict^{-1}[Route2CurrPath[\{r1\}]] \cap Route2CurrPath[\{r2\}] = \emptyset)$
typeof CurrRoute2Path: $Route2CurrPath \in RouteId \leftrightarrow PATH$

EVENTS**Initialisation****begin**

act1: $RouteId_Res := \emptyset$
act2: $RouteId_Occ := \emptyset$
act3: $Route2CurrPath := \emptyset$

end**Event** Route_Reserve **<ordinary>** $\hat{=}$ **any**

r

where

grd1: $r \notin (RouteId_Res \cup RouteId_Occ)$
grd2: $PathConflict[Route2InitPath[\{r\}]] \cap (Route2InitPath[RouteId_Res] \cup Route2CurrPath[RouteId_Occ]) = \emptyset$

then

act1: $RouteId_Res := RouteId_Res \cup \{r\}$
act2: $Route2CurrPath(r) := Route2InitPath(r)$

end**Event** Route_Cancel **<ordinary>** $\hat{=}$ **any**

r

where

grd1: $r \in RouteId_Res$
grd2: $r \notin RouteId_Occ$
grd3: $r \in dom(Route2CurrPath)$

then

act1: $RouteId_Res := RouteId_Res \setminus \{r\}$
act2: $Route2CurrPath := \{r\} \triangleleft Route2CurrPath$

end**Event** Train_Enter **<ordinary>** $\hat{=}$ **any**

r

where

grd1: $r \in RouteId_Res$
grd2: $PathConflict[Route2InitPath[\{r\}]] \cap Route2CurrPath[RouteId_Occ] = \emptyset$
grd3: $r \in dom(Route2CurrPath)$
grd4: $Route2CurrPath(r) = Route2InitPath(r)$

then

act1: $RouteId_Res := RouteId_Res \setminus \{r\}$
act2: $RouteId_Occ := RouteId_Occ \cup \{r\}$

end**Event** Train_Leave **<ordinary>** $\hat{=}$ **any**

r

where

```

    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd3:  $\text{Route2CurrPath}(r) = \text{NullPath}$ 
  then
    act1:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \setminus \{r\}$ 
    act2:  $\text{Route2CurrPath} := \{r\} \triangleleft \text{Route2CurrPath}$ 
  end
Event Train_Move  $\langle \text{ordinary} \rangle \hat{=}$ 
  any
    r
    p1
  where
    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd3:  $\text{Route2CurrPath}(r) \neq \text{NullPath}$ 
    grd4:  $p1 \in \text{PathSub}[\text{Route2CurrPath}[\{r\}]]$ 
  then
    act1:  $\text{Route2CurrPath}(r) := p1$ 
  end
END

```


MACHINE SR_M2**REFINES** SR_M1**SEES** SR2_Block_ctx**VARIABLES**

RouteId_Res
 RouteId_Occ
 Route2CurrPath
 Block2Route

INVARIANTS

inv1: $Block2Route \in BLOCK \leftrightarrow RouteId$
inv2: $\forall r1, r2. (r1 \neq r2 \wedge r1 \in (RouteId_Occ \cup RouteId_Res) \wedge r2 \in (RouteId_Occ \cup RouteId_Res)) \Rightarrow (Block2Route^{-1}[\{r1\}] \cap Block2Route^{-1}[\{r2\}]) = \emptyset$

EVENTS**Initialisation** $\langle \text{extended} \rangle$ **begin**

act1: $RouteId_Res := \emptyset$
act2: $RouteId_Occ := \emptyset$
act3: $Route2CurrPath := \emptyset$
act4: $Block2Route := \emptyset$

end**Event** Route_Reserve $\langle \text{ordinary} \rangle \hat{=}$ **extends** Route_Reserve**any** r **where**

grd1: $r \notin (RouteId_Res \cup RouteId_Occ)$
grd2: $PathConflict[Route2InitPath[\{r\}]] \cap (Route2InitPath[RouteId_Res] \cup Route2CurrPath[RouteId_Occ]) = \emptyset$
grd3: $Path2Block[Route2InitPath[\{r\}]] \cap dom(Block2Route) = \emptyset$
grd4: $Route2InitPath(r) \in dom(Path2Block)$

then

act1: $RouteId_Res := RouteId_Res \cup \{r\}$
act2: $Route2CurrPath(r) := Route2InitPath(r)$
act3: $Block2Route := Block2Route \cup \{Path2Block(Route2InitPath(r)) \mapsto r\}$

end**Event** Route_Cancel $\langle \text{ordinary} \rangle \hat{=}$ **extends** Route_Cancel**any** r **where**

grd1: $r \in RouteId_Res$
grd2: $r \notin RouteId_Occ$
grd3: $r \in dom(Route2CurrPath)$
grd4: $r \in ran(Block2Route)$

then

act1: $RouteId_Res := RouteId_Res \setminus \{r\}$
act2: $Route2CurrPath := \{r\} \triangleleft Route2CurrPath$
act3: $Block2Route := Block2Route \triangleright \{r\}$

end**Event** Train_Enter $\langle \text{ordinary} \rangle \hat{=}$ **extends** Train_Enter**any** r **where**

grd1: $r \in RouteId_Res$
grd2: $PathConflict[Route2InitPath[\{r\}]] \cap Route2CurrPath[RouteId_Occ] = \emptyset$

```

    grd3:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd4:  $\text{Route2CurrPath}(r) = \text{Route2InitPath}(r)$ 
    grd5:  $r \notin \text{RouteId\_Occ}$ 
    grd6:  $\text{Path2Block}[\text{Route2CurrPath}[\{r\}]] \cap \text{Block2Route}^{-1}[\text{RouteId\_Occ}] = \emptyset$ 
  then
    act1:  $\text{RouteId\_Res} := \text{RouteId\_Res} \setminus \{r\}$ 
    act2:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \cup \{r\}$ 
  end
Event Train_Leave  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Train_Leave
  any
     $r$ 
  where
    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd3:  $\text{Route2CurrPath}(r) = \text{NullPath}$ 
    grd4:  $\text{Block2Route} \triangleright \{r\} = \emptyset$ 
  then
    act1:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \setminus \{r\}$ 
    act2:  $\text{Route2CurrPath} := \{r\} \triangleleft \text{Route2CurrPath}$ 
  end
Event Train_Move  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Train_Move
  any
     $r$ 
     $p1$ 
     $b$ 
  where
    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd3:  $\text{Route2CurrPath}(r) \neq \text{NullPath}$ 
    grd4:  $p1 \in \text{PathSub}[\text{Route2CurrPath}[\{r\}]]$ 
    grd5:  $b \in \text{Path2Block}[\text{Route2CurrPath}[\{r\}]]$ 
    grd6:  $\text{PathReduce}(\text{Route2CurrPath}[\{r\}])(b) = p1$ 
  then
    act1:  $\text{Route2CurrPath}(r) := p1$ 
    act2:  $\text{Block2Route} := \{b\} \triangleleft \text{Block2Route}$ 
  end
END

```

MACHINE SR_M3**REFINES** SR_M2**SEES** SR3_Point_ctx**VARIABLES**

RouteId_Res
 RouteId_Occ
 Route2CurrPath
 Block2Route
 Point2Pos

INVARIANTS*inv1*: $Point2Pos \in POINT \rightarrow POS$

inv2: $\langle \text{theorem} \rangle \forall r, b \cdot r \in RouteId_Occ \wedge b \in (Path2Block[Route2CurrPath[\{r\}] \cap POINT \cap Block2Route^{-1}[\{r\}]) \Rightarrow$
 $(Point2Pos(b) = Route_Point2Pos(r)(b))$

EVENTS**Initialisation** $\langle \text{extended} \rangle$ **begin**

act1: $RouteId_Res := \emptyset$
act2: $RouteId_Occ := \emptyset$
act3: $Route2CurrPath := \emptyset$
act4: $Block2Route := \emptyset$
act5: $Point2Pos := Default_Point2Pos$

end**Event** Route_Reserve $\langle \text{ordinary} \rangle \hat{=}$ **extends** Route_Reserve**any***r***where**

grd1: $r \notin (RouteId_Res \cup RouteId_Occ)$
grd2: $PathConflict[Route2InitPath[\{r\}]] \cap (Route2InitPath[RouteId_Res] \cup Route2CurrPath[RouteId_Occ]) =$
 \emptyset
grd3: $Path2Block[Route2InitPath[\{r\}]] \cap dom(Block2Route) = \emptyset$
grd4: $Route2InitPath(r) \in dom(Path2Block)$

then

act1: $RouteId_Res := RouteId_Res \cup \{r\}$
act2: $Route2CurrPath(r) := Route2InitPath(r)$
act3: $Block2Route := Block2Route \cup \{Path2Block(Route2InitPath(r)) \mapsto r\}$

end**Event** Route_Cancel $\langle \text{ordinary} \rangle \hat{=}$ **extends** Route_Cancel**any***r***where**

grd1: $r \in RouteId_Res$
grd2: $r \notin RouteId_Occ$
grd3: $r \in dom(Route2CurrPath)$
grd4: $r \in ran(Block2Route)$

then

act1: $RouteId_Res := RouteId_Res \setminus \{r\}$
act2: $Route2CurrPath := \{r\} \triangleleft Route2CurrPath$
act3: $Block2Route := Block2Route \triangleright \{r\}$

end**Event** Point_Set $\langle \text{ordinary} \rangle \hat{=}$ **any***r**b***where**

```

    grd1:  $r \in \text{RouteId\_Res}$ 
    grd2:  $b \in (\text{Path2Block}[\text{Route2InitPath}[\{r\}]] \cap \text{POINT})$ 
    grd3:  $\text{Point2Pos}(b) \neq \text{Route\_Point2Pos}(r)(b)$ 
    grd4:  $b \in \text{Block2Route}^{-1}[\{r\}]$ 
  then
    act1:  $\text{Point2Pos}(b) := \text{Route\_Point2Pos}(r)(b)$ 
  end
Event Train_Enter  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Train_Enter
  any
     $r$ 
  where
    grd1:  $r \in \text{RouteId\_Res}$ 
    grd2:  $\text{PathConflict}[\text{Route2InitPath}[\{r\}]] \cap \text{Route2CurrPath}[\text{RouteId\_Occ}] = \emptyset$ 
    grd3:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd4:  $\text{Route2CurrPath}(r) = \text{Route2InitPath}(r)$ 
    grd5:  $r \notin \text{RouteId\_Occ}$ 
    grd6:  $\text{Path2Block}[\text{Route2CurrPath}[\{r\}]] \cap \text{Block2Route}^{-1}[\text{RouteId\_Occ}] = \emptyset$ 
    grd7:  $\forall b. b \in (\text{Path2Block}[\text{Route2InitPath}[\{r\}]] \cap \text{POINT}) \Rightarrow (\text{Point2Pos}(b) = \text{Route\_Point2Pos}(r)(b))$ 
  then
    act1:  $\text{RouteId\_Res} := \text{RouteId\_Res} \setminus \{r\}$ 
    act2:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \cup \{r\}$ 
  end
Event Train_Leave  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Train_Leave
  any
     $r$ 
  where
    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd3:  $\text{Route2CurrPath}(r) = \text{NullPath}$ 
    grd4:  $\text{Block2Route} \triangleright \{r\} = \emptyset$ 
  then
    act1:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \setminus \{r\}$ 
    act2:  $\text{Route2CurrPath} := \{r\} \triangleleft \text{Route2CurrPath}$ 
  end
Event Train_Move  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Train_Move
  any
     $r$ 
     $p1$ 
     $b$ 
  where
    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd3:  $\text{Route2CurrPath}(r) \neq \text{NullPath}$ 
    grd4:  $p1 \in \text{PathSub}[\text{Route2CurrPath}[\{r\}]]$ 
    grd5:  $b \in \text{Path2Block}[\text{Route2CurrPath}[\{r\}]]$ 
    grd6:  $\text{PathReduce}(\text{Route2CurrPath}[\{r\}]) (b) = p1$ 
  then
    act1:  $\text{Route2CurrPath}(r) := p1$ 
    act2:  $\text{Block2Route} := \{b\} \triangleleft \text{Block2Route}$ 
  end
END

```

MACHINE SR_M4**REFINES** SR_M3**SEES** SR4_TVD_ctx**VARIABLES**

RouteId_Res
 RouteId_Occ
 Route2CurrPath
 Block2Route
 Point2Pos
 Block2Status

INVARIANTS

typeof Block2Status: $Block2Status \in BLOCK \rightarrow STATUS$
inv1: $\forall b \cdot b \in dom(Block2Status) \wedge Block2Status(b) = Occupied \Rightarrow b \in dom(Block2Route)$

EVENTS**Initialisation** $\langle \text{extended} \rangle$ **begin**

act1: $RouteId_Res := \emptyset$
act2: $RouteId_Occ := \emptyset$
act3: $Route2CurrPath := \emptyset$
act4: $Block2Route := \emptyset$
act5: $Point2Pos := Default_Point2Pos$
act6: $Block2Status := BLOCK \times \{Vacant\}$

end**Event** Route_Reserve $\langle \text{ordinary} \rangle \hat{=}$ **extends** Route_Reserve**any** r **where**

grd1: $r \notin (RouteId_Res \cup RouteId_Occ)$
grd2: $PathConflict[Route2InitPath[\{r\}]] \cap (Route2InitPath[RouteId_Res] \cup Route2CurrPath[RouteId_Occ]) = \emptyset$
grd3: $Path2Block[Route2InitPath[\{r\}]] \cap dom(Block2Route) = \emptyset$
grd4: $Route2InitPath(r) \in dom(Path2Block)$

then

act1: $RouteId_Res := RouteId_Res \cup \{r\}$
act2: $Route2CurrPath(r) := Route2InitPath(r)$
act3: $Block2Route := Block2Route \cup \{Path2Block(Route2InitPath(r)) \mapsto r\}$

end**Event** Route_Cancel $\langle \text{ordinary} \rangle \hat{=}$ **extends** Route_Cancel**any** r **where**

grd1: $r \in RouteId_Res$
grd2: $r \notin RouteId_Occ$
grd3: $r \in dom(Route2CurrPath)$
grd4: $r \in ran(Block2Route)$
grd6: $Block2Status^{-1}[\{Occupied\}] \cap Block2Route^{-1}[\{r\}] = \emptyset$

then

act1: $RouteId_Res := RouteId_Res \setminus \{r\}$
act2: $Route2CurrPath := \{r\} \triangleleft Route2CurrPath$
act3: $Block2Route := Block2Route \triangleright \{r\}$

end**Event** Point_Set $\langle \text{ordinary} \rangle \hat{=}$ **extends** Point_Set**any**

```

    r
    b
  where
    grd1:  $r \in \text{RouteId\_Res}$ 
    grd2:  $b \in (\text{Path2Block}[\text{Route2InitPath}[\{r\}]] \cap \text{POINT})$ 
    grd3:  $\text{Point2Pos}(b) \neq \text{Route\_Point2Pos}(r)(b)$ 
    grd4:  $b \in \text{Block2Route}^{-1}[\{r\}]$ 
  then
    act1:  $\text{Point2Pos}(b) := \text{Route\_Point2Pos}(r)(b)$ 
  end
Event Train_Enter  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Train_Enter
  any
    r
  where
    grd1:  $r \in \text{RouteId\_Res}$ 
    grd2:  $\text{PathConflict}[\text{Route2InitPath}[\{r\}]] \cap \text{Route2CurrPath}[\text{RouteId\_Occ}] = \emptyset$ 
    grd3:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd4:  $\text{Route2CurrPath}(r) = \text{Route2InitPath}(r)$ 
    grd5:  $r \notin \text{RouteId\_Occ}$ 
    grd6:  $\text{Path2Block}[\text{Route2CurrPath}[\{r\}]] \cap \text{Block2Route}^{-1}[\text{RouteId\_Occ}] = \emptyset$ 
    grd7:  $\forall b. b \in (\text{Path2Block}[\text{Route2InitPath}[\{r\}]] \cap \text{POINT}) \Rightarrow (\text{Point2Pos}(b) = \text{Route\_Point2Pos}(r)(b))$ 
  then
    act1:  $\text{RouteId\_Res} := \text{RouteId\_Res} \setminus \{r\}$ 
    act2:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \cup \{r\}$ 
  end
Event Train_Leave  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Train_Leave
  any
    r
  where
    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd3:  $\text{Route2CurrPath}(r) = \text{NullPath}$ 
    grd4:  $\text{Block2Route} \triangleright \{r\} = \emptyset$ 
  then
    act1:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \setminus \{r\}$ 
    act2:  $\text{Route2CurrPath} := \{r\} \triangleleft \text{Route2CurrPath}$ 
  end
Event Train_Move  $\langle \text{ordinary} \rangle \hat{=}$ 
extends Train_Move
  any
    r
    p1
    b
  where
    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd3:  $\text{Route2CurrPath}(r) \neq \text{NullPath}$ 
    grd4:  $p1 \in \text{PathSub}[\text{Route2CurrPath}[\{r\}]]$ 
    grd5:  $b \in \text{Path2Block}[\text{Route2CurrPath}[\{r\}]]$ 
    grd6:  $\text{PathReduce}(\text{Route2CurrPath}[\{r\}])(b) = p1$ 
    grd7:  $\text{Block2Status}(b) = \text{Occupied}$ 
  then
    act1:  $\text{Route2CurrPath}(r) := p1$ 
    act2:  $\text{Block2Route} := \{b\} \triangleleft \text{Block2Route}$ 
    act3:  $\text{Block2Status}(b) := \text{Vacant}$ 
  end

```

```
end
Event Train_Head_Move <ordinary>  $\hat{=}$ 
  any
    r
    b
  where
    grd1:  $r \in \text{RouteId\_Occ}$ 
    grd6:  $r \in \text{dom}(\text{Route2CurrPath})$ 
    grd4:  $\text{Route2CurrPath}(r) \neq \text{NullPath}$ 
    grd2:  $b \in \text{Path2Block}[\text{Route2CurrPath}[\{r\}]]$ 
    grd3:  $b \in \text{Block2Route}^{-1}[\{r\}]$ 
    grd5:  $\text{Block2Status}(b) = \text{Vacant}$ 
  then
    act1:  $\text{Block2Status}(b) := \text{Occupied}$ 
  end
END
```

MACHINE SR_M5**REFINES** SR_M4**SEES** SR5_Signal_ctx**VARIABLES**

RouteId_Res
 RouteId_Occ
 Route2CurrPath
 Block2Route
 Point2Pos
 Block2Status
 Route2Signal

INVARIANTS

inv1: $Route2Signal \in RouteId \rightarrow ASPECT$

EVENTS**Initialisation** $\langle \text{extended} \rangle$ **begin**

act1: $RouteId_Res := \emptyset$
act2: $RouteId_Occ := \emptyset$
act3: $Route2CurrPath := \emptyset$
act4: $Block2Route := \emptyset$
act5: $Point2Pos := Default_Point2Pos$
act6: $Block2Status := BLOCK \times \{Vacant\}$
act7: $Route2Signal := RouteId \times \{Stop\}$

end**Event** Route_Reserve $\langle \text{ordinary} \rangle \hat{=}$ **extends** Route_Reserve**any***r***where**

grd1: $r \notin (RouteId_Res \cup RouteId_Occ)$
grd2: $PathConflict[Route2InitPath[\{r\}]] \cap (Route2InitPath[RouteId_Res] \cup Route2CurrPath[RouteId_Occ]) = \emptyset$
grd3: $Path2Block[Route2InitPath[\{r\}]] \cap dom(Block2Route) = \emptyset$
grd4: $Route2InitPath(r) \in dom(Path2Block)$

then

act1: $RouteId_Res := RouteId_Res \cup \{r\}$
act2: $Route2CurrPath(r) := Route2InitPath(r)$
act3: $Block2Route := Block2Route \cup \{Path2Block(Route2InitPath(r)) \mapsto r\}$

end**Event** Route_Cancel $\langle \text{ordinary} \rangle \hat{=}$ **extends** Route_Cancel**any***r***where**

grd1: $r \in RouteId_Res$
grd2: $r \notin RouteId_Occ$
grd3: $r \in dom(Route2CurrPath)$
grd4: $r \in ran(Block2Route)$
grd6: $Block2Status^{-1}[\{Occupied\}] \cap Block2Route^{-1}[\{r\}] = \emptyset$

then

act1: $RouteId_Res := RouteId_Res \setminus \{r\}$
act2: $Route2CurrPath := \{r\} \triangleleft Route2CurrPath$
act3: $Block2Route := Block2Route \triangleright \{r\}$

end**Event** Point_Set $\langle \text{ordinary} \rangle \hat{=}$ **extends** Point_Set


```

any
  r
  b
where
  grd1:  $r \in \text{RouteId\_Res}$ 
  grd2:  $b \in (\text{Path2Block}[\text{Route2InitPath}[\{r\}]] \cap \text{POINT})$ 
  grd3:  $\text{Point2Pos}(b) \neq \text{Route\_Point2Pos}(r)(b)$ 
  grd4:  $b \in \text{Block2Route}^{-1}[\{r\}]$ 
then
  act1:  $\text{Point2Pos}(b) := \text{Route\_Point2Pos}(r)(b)$ 
end
Event Train_Enter ⟨ordinary⟩  $\hat{=}$ 
extends Train_Enter
any
  r
where
  grd1:  $r \in \text{RouteId\_Res}$ 
  grd2:  $\text{PathConflict}[\text{Route2InitPath}[\{r\}]] \cap \text{Route2CurrPath}[\text{RouteId\_Occ}] = \emptyset$ 
  grd3:  $r \in \text{dom}(\text{Route2CurrPath})$ 
  grd4:  $\text{Route2CurrPath}(r) = \text{Route2InitPath}(r)$ 
  grd5:  $r \notin \text{RouteId\_Occ}$ 
  grd6:  $\text{Path2Block}[\text{Route2CurrPath}[\{r\}]] \cap \text{Block2Route}^{-1}[\text{RouteId\_Occ}] = \emptyset$ 
  grd7:  $\forall b \cdot b \in (\text{Path2Block}[\text{Route2InitPath}[\{r\}]] \cap \text{POINT}) \Rightarrow (\text{Point2Pos}(b) = \text{Route\_Point2Pos}(r)(b))$ 

  grd8:  $\text{Route2Signal}(r) = \text{Go}$ 
then
  act1:  $\text{RouteId\_Res} := \text{RouteId\_Res} \setminus \{r\}$ 
  act2:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \cup \{r\}$ 
  act3:  $\text{Route2Signal}(r) := \text{Stop}$ 
end
Event Signal_Go ⟨ordinary⟩  $\hat{=}$ 
extends Train_Enter
any
  r
where
  grd1:  $r \in \text{RouteId\_Res}$ 
  grd2:  $\text{PathConflict}[\text{Route2InitPath}[\{r\}]] \cap \text{Route2CurrPath}[\text{RouteId\_Occ}] = \emptyset$ 
  grd3:  $r \in \text{dom}(\text{Route2CurrPath})$ 
  grd4:  $\text{Route2CurrPath}(r) = \text{Route2InitPath}(r)$ 
  grd5:  $r \notin \text{RouteId\_Occ}$ 
  grd6:  $\text{Path2Block}[\text{Route2CurrPath}[\{r\}]] \cap \text{Block2Route}^{-1}[\text{RouteId\_Occ}] = \emptyset$ 
  grd7:  $\forall b \cdot b \in (\text{Path2Block}[\text{Route2InitPath}[\{r\}]] \cap \text{POINT}) \Rightarrow (\text{Point2Pos}(b) = \text{Route\_Point2Pos}(r)(b))$ 

  grd9:  $\text{Route2Signal}(r) = \text{Stop}$ 
then
  act1:  $\text{RouteId\_Res} := \text{RouteId\_Res} \setminus \{r\}$ 
  act2:  $\text{RouteId\_Occ} := \text{RouteId\_Occ} \cup \{r\}$ 
  act3:  $\text{Route2Signal}(r) := \text{Go}$ 
end
Event Train_Leave ⟨ordinary⟩  $\hat{=}$ 
extends Train_Leave
any
  r
where
  grd1:  $r \in \text{RouteId\_Occ}$ 
  grd2:  $r \in \text{dom}(\text{Route2CurrPath})$ 
  grd3:  $\text{Route2CurrPath}(r) = \text{NullPath}$ 
  grd4:  $\text{Block2Route} \triangleright \{r\} = \emptyset$ 

```

```

    then
      act1:  $RouteId\_Occ := RouteId\_Occ \setminus \{r\}$ 
      act2:  $Route2CurrPath := \{r\} \triangleleft Route2CurrPath$ 
    end
  Event Train_Move  $\langle \text{ordinary} \rangle \hat{=}$ 
  extends Train_Move
  any
     $r$ 
     $p1$ 
     $b$ 
  where
    grd1:  $r \in RouteId\_Occ$ 
    grd2:  $r \in dom(Route2CurrPath)$ 
    grd3:  $Route2CurrPath(r) \neq NullPath$ 
    grd4:  $p1 \in PathSub[Route2CurrPath[\{r\}]]$ 
    grd5:  $b \in Path2Block[Route2CurrPath[\{r\}]]$ 
    grd6:  $PathReduce(Route2CurrPath[\{r\}])(b) = p1$ 
    grd7:  $Block2Status(b) = Occupied$ 
  then
    act1:  $Route2CurrPath(r) := p1$ 
    act2:  $Block2Route := \{b\} \triangleleft Block2Route$ 
    act3:  $Block2Status(b) := Vacant$ 
  end
  Event Train_Head_Move  $\langle \text{ordinary} \rangle \hat{=}$ 
  extends Train_Head_Move
  any
     $r$ 
     $b$ 
  where
    grd1:  $r \in RouteId\_Occ$ 
    grd6:  $r \in dom(Route2CurrPath)$ 
    grd4:  $Route2CurrPath(r) \neq NullPath$ 
    grd2:  $b \in Path2Block[Route2CurrPath[\{r\}]]$ 
    grd3:  $b \in Block2Route^{-1}[\{r\}]$ 
    grd5:  $Block2Status(b) = Vacant$ 
  then
    act1:  $Block2Status(b) := Occupied$ 
  end
END

```